CLAIMS

What is claimed is:

1. A method to emulate a cellular automata based random number generator (CA-based RNG) in software, comprising:

determining emulation parameters for said CA-based RNG ;

initializing said software;

simulating behaviors of cells of said CA-based RNG in parallel; and

outputting result of said simulating step.

2. The method of claim 1, wherein said emulation parameters include at least one of a seed number of runs, and output designation.

3. The method of claim 1, wherein said step of determining emulation parameters includes retrieving said emulation parameters from at least one of a user, a file, a database, and a remote source.

4. The method of claim 1, wherein said step of determining emulation parameters includes providing default values for any emulation parameters not received from an external source.

5. The method of claim 1, wherein said simulating step includes simulating parallel site spacing.

6. The method of claim 1, wherein in said outputting step, destinations include at least one of a display, a file, a database, an application, and a remote destination.

7. A method to generate a software code which emulates a cellular automata based random number generator (CA-based RNG), said method comprising:

determining RNG parameters;

determining one or more programming language templates;

determining functional definition of said CA-based RNG;

determining initialization routines for said CA-based RNG;

determining simulation routines for said CA-based RNG;

determining simulation results destination routines for said CA-based RNG; and

outputting code for said CA-based RNG.

8. The method of claim 7, wherein said RNG parameters include at least one of an interconnection topology, a length of said desired CA-based RNG, a cellular automata truth table, one or more desired output programming language languages, one or more code output destinations, and site spacing specification.

9. The method of claim 7, wherein said step of determining RNG parameters includes retrieving said emulation parameters from at least one of one of a user, a file, a database, and a remote source.

10. The method of claim 7, wherein said step of RNG emulation parameters includes providing default values for any emulation parameters not received from an external source.

11. The method of claim 7, wherein said step of determining language templates includes retrieving said language templates from at least one of one of a user, a file, a database, and a remote source.

12. The method of claim 7, wherein said step of determining language template includes providing at least one default language template in response to no language being specified from an external source.

13. The method of claim 7, wherein said step of determining language template includes providing at least one built-in language template in response to no language being specified from an external source.

14. The method of claim 7, wherein said step of determining functional definition includes generating a Boolean sum-of-products equation for said CA-based RNG.

15. The method of claim 7, wherein said step of determining simulation routines includes generating code for one or both of operations for parallel simulation and operations for parallel site spacing.

16. The method of claim 7, wherein in said step of determining simulation results destination routines, generating code for destinations of simulation results to include at least one of a display, a file, a database, an application, and a remote destination.

17. The method of claim 7, wherein in said step of outputting code, destinations for said generated code include at least one of a display, a file, a database, a compiler, a remote destination, and an interpreter.

18. A system for generating a software code to emulate a cellular automata based random number generator (CA-based RNG), said system comprising:

a RNG-parameter-module determining RNG parameters;

a language-template-module determining language templates; and

and code-generating-module generating code for emulating said CA-based RNG.

19. The system of claim 18, wherein said RNG parameters include at least one of an interconnection topology, a length of said desired CA-based RNG, a cellular automata truth table, one or more desired output programming language languages, one or more output destinations, and site spacing specification.

20. The system of claim 18, wherein said RNG-parameter-module retrieves said emulation parameters from at least one of one of a user, a file, a database, and a remote source.

21. The system of claim 18, wherein said RNG-parameter-module provides default values for any emulation parameters not received from an external source.

22. The system of claim 18, wherein said language-template-module retrieves said language templates from at least one of one of a user, a file, a database, and a remote source.

23. The system of claim 18, wherein said language-template-module provides at least one default language template in response to no language being specified from an external source.

24. The system of claim 18, wherein said language-template-module provides at least one built-in language template in response to no language being specified from an external source.

25. The system of claim 24, wherein said code-generating-module includes:

a functional-definition-module generating a functional definition of said CA-based RNG;

an initialization-generation-module generating initialization routines;

a simulation-generation-module generating simulation routines; and

an output-generation-module generating simulation results output routines.

26. The system of claim 25, wherein said functional-definition-module (532) generates a Boolean sum-of-products equation for said CA-based RNG.

27. The system of claim 25, wherein said simulation-generation-module (536) generates code for one or both of operations for parallel simulation and operations for parallel site spacing.

28. The system of claim 25, wherein in said output destinations generated by said output-generation-module include at least one of a display, a file, a database, an application, and a remote destination.

29. The system of claim 18, wherein in said code-generating-module outputs said generated routines to at least one of a display, a file, a database, a compiler, a remote destination, and an interpreter.